NAME

chpass, **chfn**, **chsh**, **ypchpass**, **ypchfn**, **ypchsh** - add or change user database information

SYNOPSIS

chpass [**-a** *list*] [**-e** *expiretime*] [**-p** *encpass*] [**-s** *newshell*] [user]
ypchpass [**-loy**] [**-a** *list*] [**-d** *domain*] [**-e** *expiretime*] [**-h** *host*] [**-p** *encpass*] [**-s** *newshell*] [user]

DESCRIPTION

The **chpass** utility allows editing of the user database information associated with *user* or, by default, the current user.

The **chfn**, **chsh**, **ypchpass**, **ypchfn** and **ypchsh** utilities behave identically to **chpass**. (There is only one program.)

The information is formatted and supplied to an editor for changes.

Only the information that the user is allowed to change is displayed.

The options are as follows:

-a *list*   The super-user is allowed to directly supply a user database entry, in the format specified by passwd(5), as an argument. This argument must be a colon (":") separated list of all the user database fields, although they may be empty.

-e *expiretime* Change the account expire time. This option is used to set the expire time from a script as if it was done in the interactive editor.

-p *encpass*   The super-user is allowed to directly supply an encrypted password field, in the format used by crypt(3), as an argument.

-s *newshell*   Attempt to change the user's shell to *newshell*.

Possible display items are as follows:

Login:	user's login name
Password:	user's encrypted password
Uid:	user's login
Gid:	user's login group
Class:	user's general classification
Change:	password change time

Expire: account expiration time
Full Name: user's real name
Office Location: user's office location (1)
Office Phone: user's office phone (1)
Home Phone: user's home phone (1)
Other Information: any locally defined parameters for user (1)
Home Directory: user's home directory
Shell: user's login shell

NOTE(1) -   In the actual master.passwd file, these fields are comma-delimited fields
                      embedded in the FullName field.

The *login* field is the user name used to access the computer account.

The *password* field contains the encrypted form of the user's password.

The *uid* field is the number associated with the *login* field. Both of these fields should be unique across the system (and often across a group of systems) as they control file access.

While it is possible to have multiple entries with identical login names and/or identical user id's, it is usually a mistake to do so. Routines that manipulate these files will often return only one of the multiple entries, and that one by random selection.

The *gid* field is the group that the user will be placed in at login. Since BSD supports multiple groups (see groups(1)) this field currently has little special meaning. This field may be filled in with either a number or a group name (see group(5)).

The *class* field references class descriptions in */etc/login.conf* and is typically used to initialize the user's system resource limits when they login.

The *change* field is the date by which the password must be changed.

The *expire* field is the date on which the account expires.

Both the *change* and *expire* fields should be entered in the form "month day year" where *month* is the month name (the first three characters are sufficient), *day* is the day of the month, and *year* is the year.

Five fields are available for storing the user's *full name*, *office location*, *work* and *home telephone* numbers and finally *other information* which is a single comma delimited string to represent any additional gecos fields (typically used for site specific user information). Note that finger(1) will display

the office location and office phone together under the heading *Office:*.

The user's *home directory* is the full UNIX path name where the user will be placed at login.

The *shell* field is the command interpreter the user prefers. If the *shell* field is empty, the Bourne shell, */bin/sh*, is assumed. When altering a login shell, and not the super-user, the user may not change from a non-standard shell or to a non-standard shell. Non-standard is defined as a shell not found in */etc/shells*.

Once the information has been verified, **chpass** uses pwd_mkdb(8) to update the user database.

ENVIRONMENT

The vi(1) editor will be used unless the environment variable EDITOR is set to an alternate editor. When the editor terminates, the information is re-read and used to update the user database itself. Only the user, or the super-user, may edit the information associated with the user.

See pwd_mkdb(8) for an explanation of the impact of setting the PW_SCAN_BIG_IDS environment variable.

NIS INTERACTION

The **chpass** utility can also be used in conjunction with NIS, however some restrictions apply. Currently, **chpass** can only make changes to the NIS passwd maps through rpc.yppasswdd(8), which normally only permits changes to a user's password, shell and GECOS fields. Except when invoked by the super-user on the NIS master server, **chpass** (and, similarly, passwd(1)) cannot use the rpc.yppasswdd(8) server to change other user information or add new records to the NIS passwd maps. Furthermore, rpc.yppasswdd(8) requires password authentication before it will make any changes. The only user allowed to submit changes without supplying a password is the super-user on the NIS master server; all other users, including those with root privileges on NIS clients (and NIS slave servers) must enter a password. (The super-user on the NIS master is allowed to bypass these restrictions largely for convenience: a user with root access to the NIS master server already has the privileges required to make updates to the NIS maps, but editing the map source files by hand can be cumbersome.

Note: these exceptions only apply when the NIS master server is a FreeBSD system).

Consequently, except where noted, the following restrictions apply when **chpass** is used with NIS:

1. *Only the shell and GECOS information may be changed.* All other fields are restricted, even when **chpass** is invoked by the super-user. While support for changing other fields could be added, this would lead to compatibility problems with other NIS-capable systems. Even though the super-user may supply data for other fields while editing an entry, the extra information (other than the password -- see below) will be silently discarded.

Exception: the super-user on the NIS master server is permitted to change any field.

2.   *Password authentication is required.* The **chpass** utility will prompt for the user's NIS
     password before effecting any changes. If the password is invalid, all changes will be
     discarded.

     Exception: the super-user on the NIS master server is allowed to submit changes without
     supplying a password. (The super-user may choose to turn off this feature using the **-o** flag,
     described below.)

3.   *Adding new records to the local password database is discouraged.* The **chpass** utility will
     allow the administrator to add new records to the local password database while NIS is
     enabled, but this can lead to some confusion since the new records are appended to the end of
     the master password file, usually after the special NIS '+' entries. The administrator should
     use vipw(8) to modify the local password file when NIS is running.

     The super-user on the NIS master server is permitted to add new records to the NIS password
     maps, provided the rpc.yppasswdd(8) server has been started with the **-a** flag to permitted
     additions (it refuses them by default). The **chpass** utility tries to update the local password
     database by default; to update the NIS maps instead, invoke chpass with the **-y** flag.

4.   *Password changes are not permitted.* Users should use passwd(1) or yppasswd(1) to change
     their NIS passwords. The super-user is allowed to specify a new password (even though the
     "Password:" field does not show up in the editor template, the super-user may add it back by
     hand), but even the super-user must supply the user's original password otherwise
     rpc.yppasswdd(8) will refuse to update the NIS maps.

     Exception: the super-user on the NIS master server is permitted to change a user's NIS
     password with **chpass**.

There are also a few extra option flags that are available when **chpass** is compiled with NIS support:

-d *domain* Specify a particular NIS domain. The **chpass** utility uses the system domain name by
         default, as set by the domainname(1) utility. The **-d** option can be used to override a default,
         or to specify a domain when the system domain name is not set.

-h *host*   Specify the name or address of an NIS server to query. Normally, **chpass** will communicate
         with the NIS master host specified in the *master.passwd* or *passwd* maps. On hosts that have
         not been configured as NIS clients, there is no way for the program to determine this
         information unless the user provides the hostname of a server. Note that the specified

hostname need not be that of the NIS master server; the name of any server, master or slave, in a given NIS domain will do.

When using the **-d** option, the hostname defaults to "localhost". The **-h** option can be used in conjunction with the **-d** option, in which case the user-specified hostname will override the default.

-l   Force **chpass** to modify the local copy of a user's password information in the event that a user exists in both the local and NIS databases.

-o   Force the use of RPC-based updates when communicating with rpc.yppasswdd(8) ("old-mode"). When invoked by the super-user on the NIS master server, **chpass** allows unrestricted changes to the NIS passwd maps using dedicated, non-RPC-based mechanism (in this case, a UNIX domain socket). The **-o** flag can be used to force **chpass** to use the standard update mechanism instead. This option is provided mainly for testing purposes.

-y   Opposite effect of **-l**. This flag is largely redundant since **chpass** operates on NIS entries by default if NIS is enabled.

FILES
  /etc/master.passwd the user database
  /etc/passwd   a Version 7 format password file
  /etc/pw.XXXXXX temporary file
  /etc/shells   the list of approved shells

EXAMPLES
  Change the shell of the current user to '/usr/local/bin/zsh':

       chsh -s /usr/local/bin/zsh

SEE ALSO
  finger(1), login(1), passwd(1), getusershell(3), login.conf(5), passwd(5), pw(8), pwd_mkdb(8), vipw(8)

  Robert Morris and Ken Thompson, *UNIX Password security*.

HISTORY
  The **chpass** utility appeared in 4.3BSD-Reno.

BUGS
  User information should (and eventually will) be stored elsewhere.